CLAIMS

What is claimed is:

1. A data processing system comprising:

memory; and

a pointer to a location in memory, the pointer comprising:

a block field defining a block size;

a length field defining a number of blocks in a segment of memory;

an address pointing into the segment of memory; and

a finger field which denotes a block of the segment of memory into which the address points.

2. A system as claimed in claim 1 wherein the pointer to a location in memory further includes a permission field which indicates how a process may access data within the segment of memory.

3. A system as claimed in claim 2 wherein the pointer to a location in memory further comprises a capability field which identifies the pointer to a location in memory as a capability pointer having bounds and permission defined therein.

4. A system as claimed in claim 3 which bounds the segment of memory to the number of blocks indicated by the length field, each block of size 2^B where B is a value defined in the block field.

5. A system as claimed in claim 1 wherein a base address is computed from the address in the pointer to a location in memory by setting the B least significant bits of the address equal to zero, where B is the block size, and subtracting a

block index indicated by the finger field from the base bits of the address excluding the B least significant bits.

6. A system as claimed in claim 1 wherein the length field is encoded such that the number of blocks is indicated by adding a defined constant to the value in the length field for all but the smallest range of numbers and the smallest block size.

7. A system as claimed in claim 1 wherein the pointer to a location in memory further includes an increment-only bit which when set causes the system to exclude negative offsets to the address in the pointer.

8. A system as claimed in claim 7 wherein the address of the pointer to a location in memory points to the base address of a memory region within the segment, all portions of the memory segment not within the memory region having addresses less than the address in the pointer to a location in memory.

9. A system as claimed in claim 1 wherein the memory representation includes a block field, length field and finger field, respectively, for each of a segment of memory and a subsegment within the segment of memory.

10. A method of representing data in a data processing system comprising:

defining a block size in a pointer to a location in memory;

defining in the pointer to a location in memory a number of blocks in a segment of memory;

defining in the pointer to a location in memory an address pointing into the segment of memory; and

defining in the pointer to a location in memory a block of the segment of memory into which the address points.

11. A method as claimed in claim 10 further comprising indicating in the pointer to a location in memory how a process may access data within the segment of memory.

12. A method as claimed in claim 11 further comprising identifying within a pointer to a location in memory that the representation is a capability pointer having bounds and permission defined therein.

13. A method as claimed in claim 12 wherein the pointer to a location in memory defines a number of blocks indicated by a length field, each block of size 2^B where B is a value defined in a block field.

14. A method as claimed in claim 10 further comprising computing a base address from the address in the pointer to a location in memory by setting the B least significant bits of the address equal to zero, where B is the block size, and subtracting a block index from the base bits of the address excluding the B least significant bits.

15. A method as claimed in claim 10 further comprising determining the number of blocks defined by the memory representation by adding a defined constant to a value in a length field for all but the smallest range of numbers and smallest block size.

16. A method as claimed in claim 10 further comprising excluding negative offsets to the address in the memory representation where an increment-only bit is included in the representation.

17. A method as claimed in claim 16 wherein the address of the pointer to a location in memory points to the base address of a memory region within the segment, all

portions of the memory segment not within the memory region having addresses less than the address in the pointer to a location in memory.

18. A method as claimed in claim 10 wherein the memory representation includes a block field, length field and finger field, respectively, for each of a segment of memory and a subsegment within the segment of memory.

19. A computer program product comprising:

a computer usable medium for storing data; and

a set of computer program instructions embodied on the computer usable medium, including a pointer to a location in memory, the pointer comprising:

a block field defining a block size;

a length field defining a number of blocks in a segment of memory;

an address pointing into the segment of memory; and

a finger field which denotes a block of the segment of memory into which the address points.

20. A computer program product as claimed in claim 19 wherein the pointer to a location in memory further includes a permission field which indicates how a process may access data within the segment of memory.

21. A computer program product as claimed in claim 20 wherein the pointer to a location in memory further comprises a capability field which identifies the pointer to a location in memory as a capability pointer having bounds and permission defined therein.

22. A computer program product as claimed in claim 21 which bounds the segment of memory to the number of blocks indicated by the length field, each block of size 2^B where B is a value defined in the block field.

23. A computer program product as claimed in claim 19 wherein a base address is computed from the address in the pointer to a location in memory by setting the B least significant bits of the address equal to zero, where B is the block size, and subtracting a block index indicated by the finger field from the base bits of the address excluding the B least significant bits.

24. A computer program product as claimed in claim 19 wherein the length field is encoded such that the number of blocks is indicated by adding a defined constant to the value in the length field for all but the smallest range of numbers and the smallest block size.

25. A computer program product system as claimed in claim 19 wherein the pointer to a location in memory further includes an increment-only bit which when set causes the system to exclude negative offsets to the address in the pointer to a location in memory.

26. A computer program product as claimed in claim 25 wherein the address of the pointer to a location in memory points to the base address of a memory region within the segment, all portions of the memory segment not within the memory region having addresses less than the address in the pointer to a location in memory.

27. A computer program product as claimed in claim 19 wherein the memory representation includes a block field, length field and finger field, respectively,

for each of a segment of memory and a subsegment within the segment of memory.

28. A computer data signal comprising a pointer to a location in memory, the pointer comprising:

a block field defining a block size;

a length field defining a number of blocks in a segment of memory;

an address pointing into the segment of memory; and

a finger field which denotes a block of the segment of memory into which the address points.

29. A computer data signal as claimed in claim 28 wherein the pointer to a location in memory further includes a permission field which indicates how a process may access data within the segment of memory.

30. A computer data signal as claimed in claim 29 wherein the pointer to a location in memory further comprises a capability field which identifies the pointer to a location in memory as a capability pointer having bounds and permission defined therein.

31. A computer data signal as claimed in claim 30 which bounds the segment of memory to the number of blocks indicated by the length field, each block of size 2^B where B is a value defined in the block field.

32. A computer data signal as claimed in claim 28 wherein a base address is computed from the address in the pointer to a location in memory by setting the B least significant bits of the address equal to zero, where B is the block size, and subtracting a block index indicated by the finger field from the base bits of the address excluding the B least significant bits.

33. A computer data signal as claimed in claim 28 wherein the length field is encoded such that the number of blocks is indicated by adding a defined constant to the value in the length field for all but the smallest range of numbers and the smallest block size.

34. A computer data signal as claimed in claim 28 wherein the pointer to a location in memory further includes an increment-only bit which when set causes the system to exclude negative offsets to the address in the pointer to a location in memory.

35. A computer data signal as claimed in claim 34 wherein the address of the pointer to a location in memory points to the base address of a memory region within the segment, all portions of the memory segment not within the memory region having addresses less than the address in the pointer to a location in memory.

36. A computer data signal as claimed in claim 28 wherein the memory representation includes a block field, length field and finger field, respectively, for each of a segment of memory and a subsegment within the segment of memory.